USB to SPI Bridging with Microchip USB4715 and USB49xx Hubs

Author: Jeffrey Hunt
Microchip Technology, Inc.

INTRODUCTION

The USB to SPI bridging feature gives system designers using Microchip hubs expanded system control and potential BOM reduction. The use of a separate USB to SPI device is no longer required and a downstream USB port is not lost as occurs when a standalone USB to SPI device is implemented. This feature is available on Microchip hubs which contain the internal Hub Feature Controller and a SPI Interface. These hubs include USB4914, USB4916, USB4925 USB4927 and USB4715.

Commands may be sent from the USB Host to the internal Hub Feature Controller device in the Microchip hub to perform the following functions:

- Enable SPI Pass-Through Interface
- SPI Write/Read
- Disable SPI Pass-Through Interface

SECTIONS

REFERENCES

Consult the following documents for details on the specific parts referred to in this document.

- *USB4914 Data Sheet*
- *USB4916 Data Sheet*
- *USB4925 Data Sheet*
- *USB4927 Data Sheet*
- *USB4715 Data Sheet*
- *Configuration of the USB49x4 Application Note*

1.0 GENERAL INFORMATION

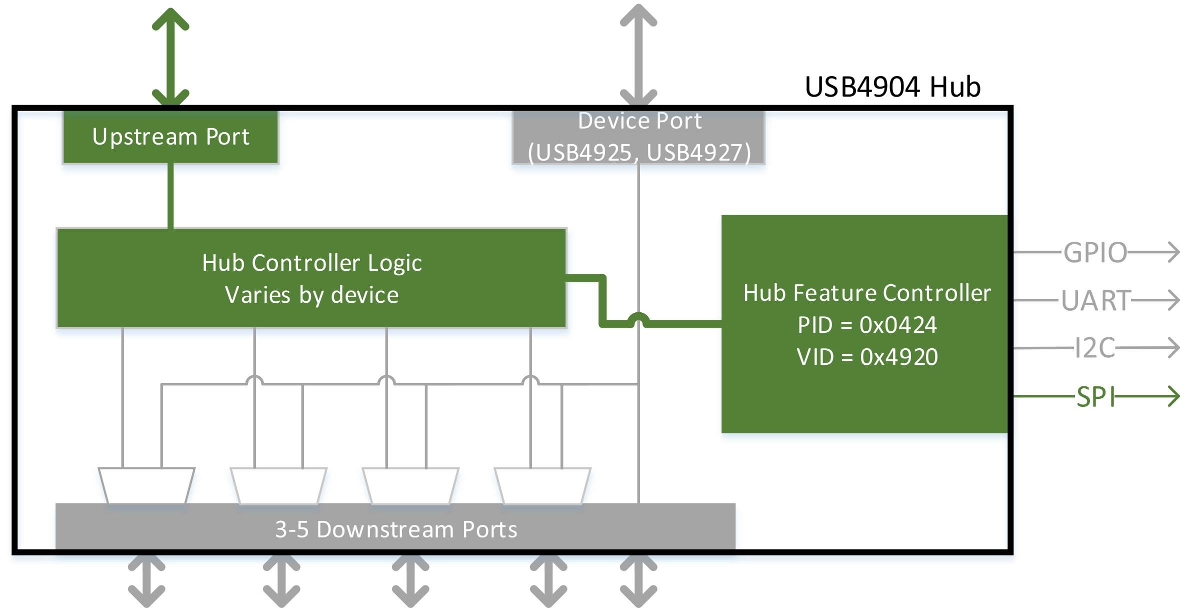
The USB Bridging features in Microchip hubs work via host commands sent to a Hub Feature Controller embedded within the hub located on an additional internal USB port. In order for the bridging features to work correctly, this internal Hub Feature Controller must be enabled by default. See the table below for details on default Hub Feature Controller settings by part number. The SPI interface is always enabled after reset. It can be disabled by setting the SPI_MASTER_DIS bit in the PAD_MUX_CTL register.

TABLE 1: DEFAULT SETTINGS FOR HUB FEATURE CONTROLLER ENABLE

Part Number	Part Summary	Hub Controller Default Setting
USB4914	One USB Upstream port, Two USB CarPlay ports, 1 standard USB port non-removable	Enabled
USB4916	One USB Upstream port, Four USB CarPlay ports, 1 standard USB port non-removable	Enabled
USB4925	One USB upstream port, one secondary USB downstream port, Two USB CarPlay ports, 1 standard USB port non-removable.	Enabled
USB4927	One USB upstream port, one secondary USB downstream port, Four USB CarPlay ports, 1 standard USB port non-removable.	Enabled
USB4715	One USB upstream port, Four USB Flex ports	Enabled

The Hub Feature Controller is connected to an extra internal port in the hub. For example, in a four port hub, the Hub Feature Controller is connected to port 5. The Product ID (PID) for the Hub Feature Controller is 0x4904. All bridging commands are addressed to the Hub Feature Controller, not the Hub.

FIGURE 1: MICROCHIP HUB FEATURE CONTROLLER BLOCK DIAGRAM



1.1 SPI Bridging Commands

The following SPI Functions are supported:

- Enable the SPI Pass-Through Interface
- SPI Write/(READ)
- Disable the SPI Pass-Through Interface

1.1.1 ENABLE THE SPI PASS-THROUGH INTERFACE

To acquire the SPI interface, the host must send a CMD_SPI_ENTER_PASSTHRU setup packet. before performing any SPI Write or Read commands. The SPI interface may operate at either 30MHz or 60HHz (based on pin-strapping).

1.1.2 SPI WRITE/(READ)

The API interface works as a complete pass-through. This means that the host must properly arrange data payloads in the appropriate SPI compatible format and bit order, including the SPI slave device address. Up to 512 Bytes of data payload may be sent per SPI write command sequence.

Data may also be read from a SPI device via the SPI Write command. Up to 512 bytes of data may be read from the SPI device per read. The read data is stored in the internal registers of the hub starting at register 0xBFD23310. To retrieve the data, you must use a USB to Register Read command. Further details can be found in *Configuration of the USB4914 / USB4916 / USB4925 / USB4927 / USB4715 Application Note.*

1.1.3 DISABLE THE SPI PASS-THROUGH INTERFACE

The SPI interface may be disabled after writing/reading to the device is complete.

1.2 SPI Interface Setup Requirements

1.2.1 SPI MASTER INTERFACE

The SPI Interface always acts as a SPI master.

1.2.2 SELECTING SPI FREQUENCY

The SPI Interface may operate at either 30MHz or 60MHz. The speed is selected by pin strapping the SPI_SD_SEL pin (which is the SPI_DO pin during runtime) and is detected at power-on or at the end of reset. The strapping options are:

- GND (Logical 0) = 30MHz
- 3.3V (Logical 1) = 60MHz

When running in 30MHz mode, the SPI_DO/SPI_SPD_SEL pin will drive to 0V during a suspend state. When running in 60 MHz mode, the SPI_DO/SPI_SPD_SEL pin will tri-state during a suspend state.

1.2.3 SPI MODES OF OPERATION

Both SPI Modes 0 and 3 are supported:

- Mode 0: Clock Polarity = 0, Clock Edge = 1
- Mode 3: Clock Polarity = 1, Clock Edge = 0

Dual Output Enable mode is also supported.

The default mode of operation is Mode 0 with Dual Output Enable mode disabled. If the mode of operation is to be modified, a register write to the SPI_CONTROL register must be performed. See *Configuration of the USB4914 / USB4916 / USB4925 / USB4927 / USB4715 Application Note* for details on how to modify that register.

2.0 PART NUMBER SPECIFIC INFORMATION

2.1 Part Summary

In USB4904 devices, SPI interface signals are associated with dedicated pins.

TABLE 2: USB3613 AND USB3813 SPI INTERFACE PIN NUMBERS

SKU	SPI_DI	SPI_CE_EN	SPI_DO	SPI_CLK	SPI_D2	SPI_D3
USB4914	23	24	22	21	26	27
USB4916	31	32	30	29	34	35
USB4925	23	24	22	21	26	27
USB4927	31	32	30	29	34	35
USB4715	23	24	22	21	26	27
Notes:		This pin is also the SPI_SPD_-SEL Strap which selects between 30 MHz and 60 MHz speed of operation.				

3.0 MPLABCC DLL IMPLEMENTATION

The simplest method for implementing the USB to SPI bridging functions is to use the publicly available MPLABCC DLL. MPLABCC is available for Windows and Linux operating systems. Visit the product page for any of the hubs listed in this document on microchip.com to download the MPLABCC package for the desired Operating System. Using the libraries available in MPLABCC, the bridging features can be implemented in the C programming language.

The MPLABCC package contains the following:

- User’s Manual: Detail description of how to use the DLL API to call each function
- Release Notes:
- Library Files:
  - For Windows: A “`.dll`” and a “`.lib`” file
  - For Linux: a “`.cpp`” file that can be built into a “`.a`” file
- Example code

3.1 Commands included in the SDK

- `MchpUsbSpiSetConfig`: This API enables/disables the SPI interface.
- `MchpUsbSpiFlashWrite`: This API performs write opeartion to SPI Flash memory..
- `MchpUsbSpiFlashRead`: This API performs read operation from SPI Flash memory.
- `MchpUsbSpiTransfer`: This API performs write operation to the SPI Interface. SPI transfer to/from arbitrary SPI slaves is not supported.

For additional details on how to use the MPLABCC DLL to implement USB to SPI bridging, download the MPLABCC package and refer to the User’s Manual.

4.0 MANUAL IMPLEMENTATION

The USB to SPI bridging features may be implemented at the lowest level if you have the ability to build USB packets. This approach is required if you are not using a Windows or Linux host system and cannot use the SDK.

The details of the SPI pass-through control packets are shown below.

4.1 Enable SPI Pass-Through Interface Command

The following SETUP packet command is required to enable the SPI pass-through interface. The interface must be enabled before any Write or Read commands may be performed. Note that there is no data phase to this USB transaction,

TABLE 3: USB SETUP COMMAND

Setup Parameter	Value	Description
bmRequestType	0x41	Vendor specific command, Host-to-device data transfer
bRequest	0x60	Register read command: `CMD_SPI_ENTER_PASSTHRU`
wValue	0x0000	Reserved
wIndex	0x0000	Reserved
wLength	0x00	No data stage

4.2 SPI Write Command

This command is used to write data to or read data from a SPI peripheral connected to the USB hub.

TABLE 4: USB SETUP COMMAND

Setup Parameter	Value	Description
bmRequestType	0x41	Vendor specific command, Host-to-device data transfer
bRequest	0x61	Register read command: `CMD_SPI_WRITE`
wValue	0xXXXX	The total length of data to be sent to the SPI peripheral (the size of the data following the SETUP packet).
wIndex	0x0000	Reserved
wLength	0xNN	The number of bytes the SPI interface will return for the command sent

The maximum amount of data that can be read from one USB command is 512 Bytes by specifying wValue = 517 and wLength = 5.

The maximum amount of data that can be written to a SPI peripheral is 256 Bytes by specifying wValue = wLength = 260.

4.2.1 SPI WRITE USB TRANSACTION SEQUENCE:

1. SETUP PACKET: To send ‘Write Enable’ OpCode to SPI ROM (wValue = wLegth = 1)
2. DATA: 0x06 (opcode for ‘Write Enable).
3. STATUS: An IN-Zero Length Packet is sent from hub.
4. SETUP PACKET: To send data payload
5. DATA: EP0 Data to SPI ROM with 0x02 + 24bit SPI address + Data Stream
6. STATUS: If an IN-Zero Length Packet is sent from hub, transfer was a success. If an IN-STALL packet is sent from hub, there was an error during transfer.

4.2.2 SPI READ USB TRANSACTION SEQUENCE:

1.) SETUP PACKET: As defined above.

2.) DATA: EP0 OUT Data to SPI ROM with 0x0B + 24Bit SPI Address + 0x00 (dummy byte).

3.) STATUS: If an IN-Zero Length Packet is sent from hub, transfer was a success. If an IN-STALL packet is sent from hub, there was an error during transfer.

4.) Perform Configuration Register Read on hub starting at register 0x4A10 to retrieve read data. See *AN 26.18 Configuration of the USB253x / USB3x13 / USB46x4*

4.3 Disable SPI Pass-Through Interface Command

The follow SETUP packet command is required to enable the SPI pass-through interface. Note that there is no data phase to this USB transaction,

TABLE 5: USB SETUP COMMAND

Setup Parameter	Value	Description
bmRequestType	0x41	Vendor specific command, Host-to-device data transfer
bRequest	0x62	Register read command: `CMD_SPI_ENTER_PASSTHRU`
wValue	0x0000	Reserved
wIndex	0x0000	Reserved
wLength	0x00	No data stage

5.0 EXAMPLES

5.1 Write 512 Bytes to a SPI ROM

1. Enable the SPI Pass-Through Interface

TABLE 6: ENABLE SPI INTERFACE SETUP COMMAND

Setup Parameter	Value
bmRequestType	0x41
bRequest	0x60
wValue	0x0000
wIndex	0x0000
wLength	0x00

2. Send a SPI Write/Read Command to read 512 Bytes of data.

TABLE 7: SPI WRITE SETUP COMMAND

Setup Parameter	Value
bmRequestType	0x41
bRequest	0x61
wValue	0x0205 (517)
wIndex	0x0000
wLength	0x0005

3. EP0 OUT data = 0x0B, 0xXX, 0xYY, 0xZZ, 0x00, 0xXX, 0xYY, 0xZZ.

Note: 0xXX, 0xYY, 0xZZ is the 24 Bit physical SPI address of the SPI peripheral.

4. Read response via USB to Configuration Register Read from register 0x4A10. Further details can be found in *AN 26.18 Configuration of the USB253x / USB3x13 / USB46x4.*
5. Do one of the following:
   - Close the SPI Interface with the Disable SPI Pass-Through Command
   - Wait for the manufacturer specified time before performing another Read/Write command.
   - Send RDSR commands until the BUSY field is cleared before performing another Read/Write command.

6.0 EXAMPLES

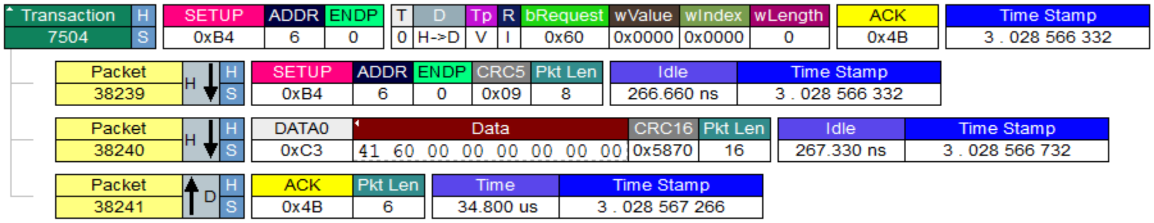
6.1 Enable the SPI Pass-Through Interface

1. **Command Phase (SETUP Transaction):** Send the following SETUP Register Read Command to Endpoint 0 of the Hub Feature Controller to enable the SPI pass-through interface:

TABLE 8: EXAMPLE SPI WRITE SETUP PACKET

Field	Value	Note
bmRequestType	0x41	—
bRequest	0x60	—
wValue	0x0000	—
wIndex	0x0000	—
wLength	0x0000	—

FIGURE 2: SETUP TRANSACTION EXAMPLE



Transaction	H S	SETUP	ADDR	ENDP	T	D	Tp	R	bRequest	wValue	wIndex	wLength	ACK	Time Stamp
7504		0xB4	6	0	0	H->D	V	I	0x60	0x0000	0x0000	0	0x4B	3 . 028 566 332

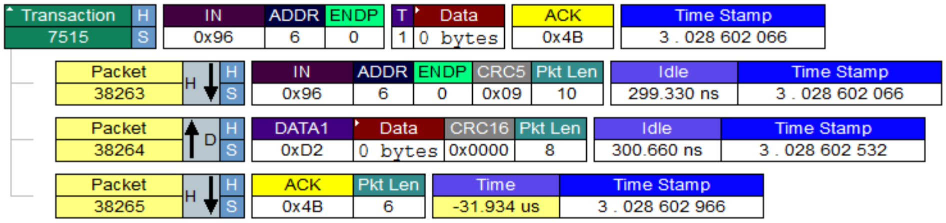
Packet	H	H S	SETUP	ADDR	ENDP	CRC5	Pkt Len	Idle	Time Stamp
38239	↓		0xB4	6	0	0x09	8	266.660 ns	3 . 028 566 332

Packet	H	H S	DATA0	Data	CRC16	Pkt Len	Idle	Time Stamp
38240	↓		0xC3	41 60 00 00 00 00 00 00	0x5870	16	267.330 ns	3 . 028 566 732

Packet	D	H S	ACK	Pkt Len	Time	Time Stamp
38241	↑		0x4B	6	34.800 us	3 . 028 567 266

2. **Status (IN Transaction):** The host sends an IN packet to the Hub Feature Controller, to which the Hub Feature controller replies with a zero data length packet. The host ACKs to complete the bridging command.

FIGURE 3: IN TRANSACTION EXAMPLE



Transaction	H S	IN	ADDR	ENDP	T	Data	ACK	Time Stamp
7515		0x96	6	0	1	0 bytes	0x4B	3 . 028 602 066

Packet	H	H S	IN	ADDR	ENDP	CRC5	Pkt Len	Idle	Time Stamp
38263	↓		0x96	6	0	0x09	10	299.330 ns	3 . 028 602 066

Packet	D	H S	DATA1	Data	CRC16	Pkt Len	Idle	Time Stamp
38264	↑		0xD2	0 bytes	0x0000	8	300.660 ns	3 . 028 602 532

Packet	H	H S	ACK	Pkt Len	Time	Time Stamp
38265	↓		0x4B	6	-31.934 us	3 . 028 602 966

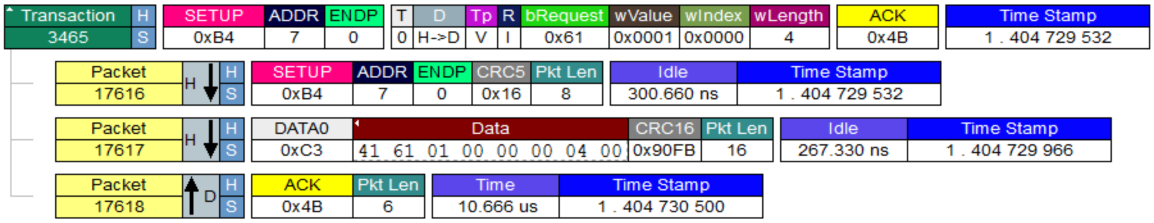
6.2 Read the JEDEC ID from an attached SPI Device

1. **Command Phase 1 (SETUP Transaction 1):** The JEDEC ID gives manufacturer information and memory information. This example shows how to read the JEDEC ID from an attached SPI device. Send the following SETUP Register Read Command to Endpoint 0 of the Hub Feature Controller send a SPI Write command to the attached SPI device

TABLE 9: SPI JEDEC ID READ SETUP PACKET EXAMPLE

Field	Value	Note
bmRequestType	0x41	—
bRequest	0x61	—
wValue	0x0001	—
wIndex	0x0000	—
wLength	0x0004	—

FIGURE 4: SPI JEDEC ID READ SETUP TRANSACTION EXAMPLE



Transaction	H	SETUP	ADDR	ENDP	T	D	Tp	R	bRequest	wValue	wIndex	wLength	ACK	Time Stamp
3465	S	0xB4	7	0	0	H->D	V	I	0x61	0x0001	0x0000	4	0x4B	1 . 404 729 532

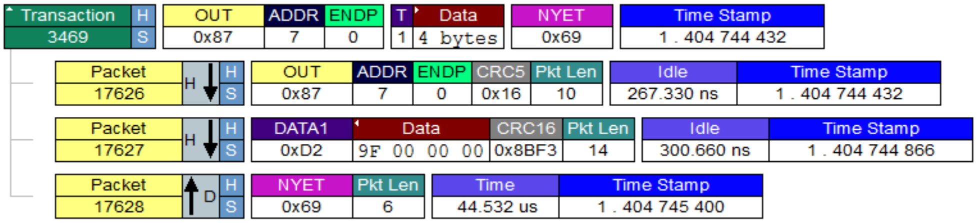
Packet		H	SETUP	ADDR	ENDP	CRC5	Pkt Len	Idle	Time Stamp
17616	H ↓	S	0xB4	7	0	0x16	8	300.660 ns	1 . 404 729 532

Packet		H	DATA0	Data	CRC16	Pkt Len	Idle	Time Stamp
17617	H ↓	S	0xC3	41 61 01 00 00 00 04 00	0x90FB	16	267.330 ns	1 . 404 729 966

Packet		H	ACK	Pkt Len	Time	Time Stamp
17618	↑ D	S	0x4B	6	10.666 us	1 . 404 730 500

2. **Data Phase 1 (OUT Transaction 1):** Host sends an OUT packet followed by the data bytes of length wLength. In this example, the 0x9F data is the opcode for reading a JEDEC ID in a SPI device. Hub Feature Controller responds with a NYET after receiving the data.

FIGURE 5: SPI JEDEC ID IN TRANSACTION EXAMPLE



Transaction	H	OUT	ADDR	ENDP	T	Data	NYET	Time Stamp
3469	S	0x87	7	0	1	4 bytes	0x69	1 . 404 744 432

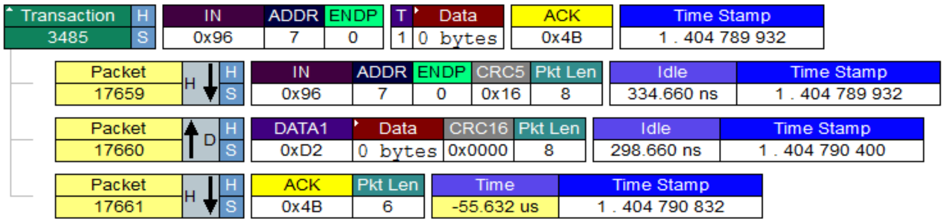
Packet		H	OUT	ADDR	ENDP	CRC5	Pkt Len	Idle	Time Stamp
17626	H ↓	S	0x87	7	0	0x16	10	267.330 ns	1 . 404 744 432

Packet		H	DATA1	Data	CRC16	Pkt Len	Idle	Time Stamp
17627	H ↓	S	0xD2	9F 00 00 00	0x8BF3	14	300.660 ns	1 . 404 744 866

Packet		H	NYET	Pkt Len	Time	Time Stamp
17628	↑ D	S	0x69	6	44.532 us	1 . 404 745 400

3. **Status Phase 1 (IN Transaction 1):** Host sends an IN packet to complete the USB Transfer. Hub Feature Controller responds with a zero length data packet. The host ACKs to complete the bridging command.

FIGURE 6: SPI JEDEC ID OUT TRANSACTION EXAMPLE



Transaction	H	IN	ADDR	ENDP	T	Data	ACK	Time Stamp
3485	S	0x96	7	0	1	0 bytes	0x4B	1 . 404 789 932

Packet		H	IN	ADDR	ENDP	CRC5	Pkt Len	Idle	Time Stamp
17659	H ↓	S	0x96	7	0	0x16	8	334.660 ns	1 . 404 789 932

Packet		H	DATA1	Data	CRC16	Pkt Len	Idle	Time Stamp
17660	↑ D	S	0xD2	0 bytes	0x0000	8	298.660 ns	1 . 404 790 400

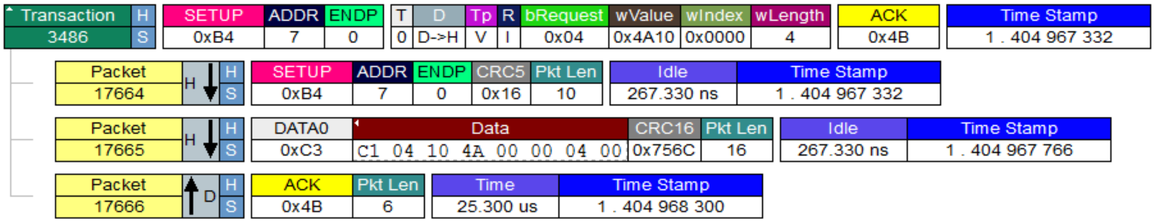
Packet		H	ACK	Pkt Len	Time	Time Stamp
17661	H ↓	S	0x4B	6	-55.632 us	1 . 404 790 832

4. **Retrieve the Returned Data (SETUP Transaction 2):** The SPI device will respond to the opcode command and the returned data will be stored in the hub's internal register starting at address 0x4A10. A USB to Register Read command can retrieve the data. This command is setup as:

TABLE 10: SPI JEDEC ID REGISTER READ SETUP PACKET EXAMPLE

Field	Value	Note
bmRequestType	0xC1	—
bRequest	0x04	—
wValue	0x4A10	The hub's internal register address.
wIndex	0x0000	—
wLength	0x0004	A JEDEC ID request will return 4 bytes. The first byte will be a dummy 0x00.

FIGURE 7: SPI JEDEC ID REGISTER READ SETUP TRANSACTION EXAMPLE



Transaction	H S	SETUP	ADDR	ENDP	T	D	Tp	R	bRequest	wValue	wIndex	wLength	ACK	Time Stamp
3486		0xB4	7	0	0	D->H	V	I	0x04	0x4A10	0x0000	4	0x4B	1 . 404 967 332

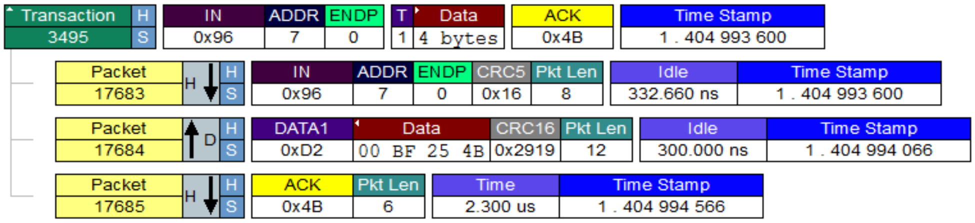
Packet		H S	SETUP	ADDR	ENDP	CRC5	Pkt Len	Idle	Time Stamp
17664	H ↓		0xB4	7	0	0x16	10	267.330 ns	1 . 404 967 332

Packet		H S	DATA0	Data	CRC16	Pkt Len	Idle	Time Stamp
17665	H ↓		0xC3	C1 04 10 4A 00 00 04 00	0x756C	16	267.330 ns	1 . 404 967 766

Packet		H S	ACK	Pkt Len	Time	Time Stamp
17666	↑ D		0x4B	6	25.300 us	1 . 404 968 300

5. **Data Phase 2 (IN Transaction 2):** Host sends an IN packet to retrieve the data from the 0x4A10 register. In this example, the JEDEC ID that is returned is 0xBF, 0x25, 0x4B. The host replies with an ACK after receiving the data.

FIGURE 8: SPI JEDEC ID REGISTER READ IN TRANSACTION EXAMPLE



Transaction	H S	IN	ADDR	ENDP	T	Data	ACK	Time Stamp
3495		0x96	7	0	1	4 bytes	0x4B	1 . 404 993 600

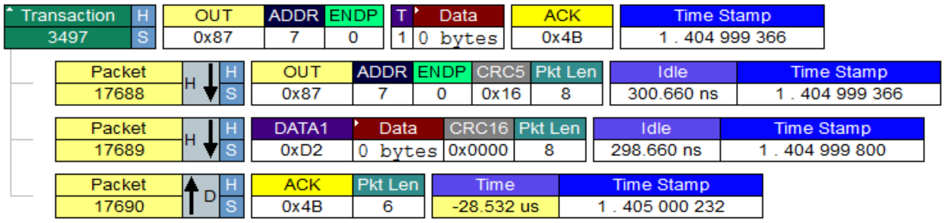
Packet		H S	IN	ADDR	ENDP	CRC5	Pkt Len	Idle	Time Stamp
17683	H ↓		0x96	7	0	0x16	8	332.660 ns	1 . 404 993 600

Packet		H S	DATA1	Data	CRC16	Pkt Len	Idle	Time Stamp
17684	↑ D		0xD2	00 BF 25 4B	0x2919	12	300.000 ns	1 . 404 994 066

Packet		H S	ACK	Pkt Len	Time	Time Stamp
17685	H ↓		0x4B	6	2.300 us	1 . 404 994 566

6. **Status Phase 2 (OUT Transaction 2):** Host sends an OUT packet followed by a zero length data packet. The Hub Feature Controller ACKs to complete the bridging command.

FIGURE 9: SPI JEDEC ID REGISTER READ OUT TRANSACTION EXAMPLE



Transaction	H S	OUT	ADDR	ENDP	T	Data	ACK	Time Stamp
3497		0x87	7	0	1	0 bytes	0x4B	1 . 404 999 366

Packet		H S	OUT	ADDR	ENDP	CRC5	Pkt Len	Idle	Time Stamp
17688	H ↓		0x87	7	0	0x16	8	300.660 ns	1 . 404 999 366

Packet		H S	DATA1	Data	CRC16	Pkt Len	Idle	Time Stamp
17689	H ↓		0xD2	0 bytes	0x0000	8	298.660 ns	1 . 404 999 800

Packet		H S	ACK	Pkt Len	Time	Time Stamp
17690	↑ D		0x4B	6	-28.532 us	1 . 405 000 232

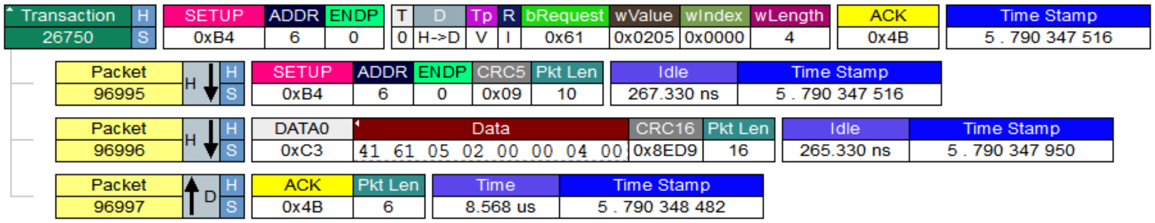
6.3 Read 512 Bytes from an attached SPI Device

1. **Command Phase 1 (SETUP Transaction 1):** This example shows how to perform a block read of 512 bytes (the maximum number of bytes per command) from an attached SPI device. Send the following SETUP Register Read Command to Endpoint 0 of the Hub Feature Controller send a SPI Write command to the attached SPI device

TABLE 11: SPI BLOCK READ COMMAND SETUP PACKET EXAMPLE

Field	Value	Note
bmRequestType	0x41	—
bRequest	0x61	—
wValue	0x0205 (517)	The first 5 Bytes of any SPI read must be ignored; Therefore, 5 must be added to the number of Bytes to be read.
wIndex	0x0000	—
wLength	0x0004	—

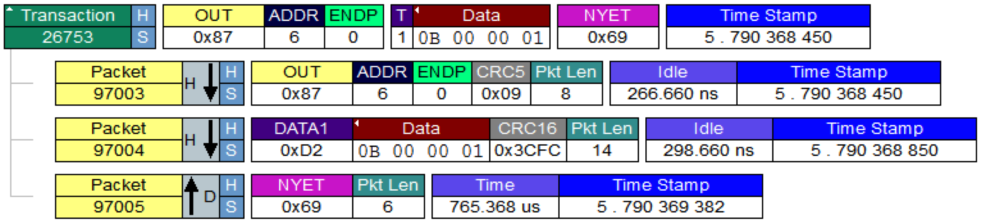
FIGURE 10: SPI BLOCK READ COMMAND SETUP TRANSACTION EXAMPLE



```
Transaction  H  SETUP  ADDR  ENDP  T  D     Tp  R  bRequest  wValue  wIndex  wLength  ACK   Time Stamp
26750        S  0xB4   6     0     0  H->D  V   I  0x61      0x0205  0x0000  4        0x4B  5 . 790 347 516

Packet  H  H  SETUP  ADDR  ENDP  CRC5  Pkt Len  Idle        Time Stamp
96995      S  0xB4   6     0     0x09  10       267.330 ns  5 . 790 347 516

Packet  H  H  DATA0  Data                       CRC16   Pkt Len  Idle        Time Stamp
96996      S  0xC3   41 61 05 02 00 00 04 00    0x8ED9  16       265.330 ns  5 . 790 347 950

Packet  D  H  ACK   Pkt Len  Time      Time Stamp
96997      S  0x4B  6        8.568 us  5 . 790 348 482
```

2. **Data Phase 1 (OUT Transaction 1):** Host sends an OUT packet followed by the data bytes of length wLength. In this example, the 0x0B is the opcode for a block read in this specific SPI device, the next 0x00,0x00,0x01 is the register address to begin reading from the SPI device. The Hub Feature Controller responds with a NYET after receiving the data.

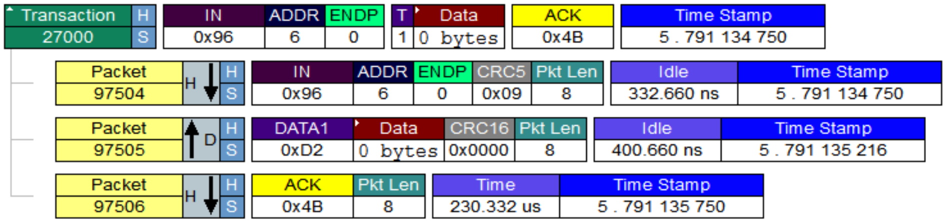
FIGURE 11: SPI BLOCK READ COMMAND IN TRANSACTION EXAMPLE



```
Transaction  H  OUT   ADDR  ENDP  T  Data         NYET  Time Stamp
26753        S  0x87  6     0     1  0B 00 00 01  0x69  5 . 790 368 450

Packet  H  H  OUT   ADDR  ENDP  CRC5  Pkt Len  Idle        Time Stamp
97003      S  0x87  6     0     0x09  8        266.660 ns  5 . 790 368 450

Packet  H  H  DATA1  Data         CRC16   Pkt Len  Idle        Time Stamp
97004      S  0xD2   0B 00 00 01  0x3CFC  14       298.660 ns  5 . 790 368 850

Packet  D  H  NYET  Pkt Len  Time        Time Stamp
97005      S  0x69  6        765.368 us  5 . 790 369 382
```

3. **Status Phase 1 (IN Transaction 1):** Host sends an IN packet to complete the USB Transfer. Hub Feature Controller responds with a zero length data packet. The host ACKs to complete the bridging command.

FIGURE 12: SPI BLOCK READ COMMAND OUT TRANSACTION EXAMPLE



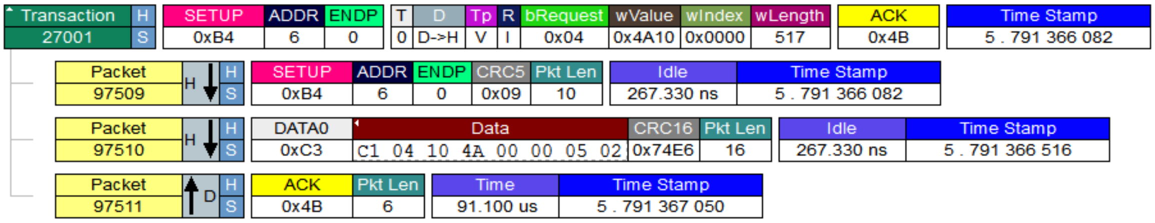
```
Transaction  H  IN    ADDR  ENDP  T  Data     ACK   Time Stamp
27000        S  0x96  6     0     1  0 bytes  0x4B  5 . 791 134 750

Packet  H  H  IN    ADDR  ENDP  CRC5  Pkt Len  Idle        Time Stamp
97504      S  0x96  6     0     0x09  8        332.660 ns  5 . 791 134 750

Packet  D  H  DATA1  Data     CRC16   Pkt Len  Idle        Time Stamp
97505      S  0xD2   0 bytes  0x0000  8        400.660 ns  5 . 791 135 216

Packet  H  H  ACK   Pkt Len  Time        Time Stamp
97506      S  0x4B  8        230.332 us  5 . 791 135 750
```

4. **Retrieve the Returned Data (SETUP Transaction 2):** The SPI device will respond to the opcode command and the returned data will be stored in the hub's internal register starting at address 0x4A10. For a SPI Block Read, the first 5 bytes must be ignored. A USB to Register Read command can retreive the data. This command is setup as:

TABLE 12: REGISTER READ SETUP PACKET EXAMPLE

Field	Value	Note
bmRequestType	0xC1	—
bRequest	0x04	—
wValue	0x4A10	The hub's internal register address
wIndex	0x0000	—
wLength	0x0205	517 Bytes will be read (512 + 5 dummy bytes)

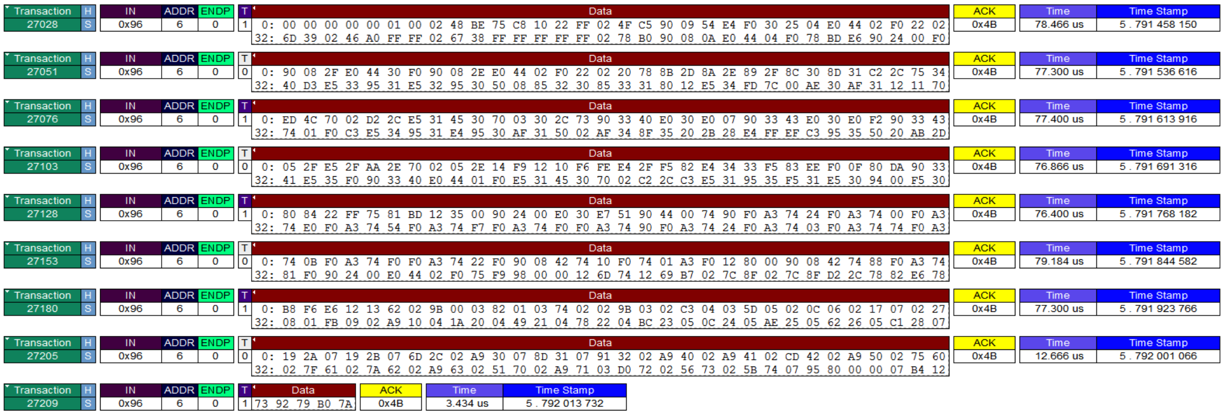
FIGURE 13: REGISTER READ SETUP TRANSACTION EXAMPLE



```
Transaction H   SETUP ADDR ENDP   T  D    Tp R  bRequest wValue wIndex wLength   ACK    Time Stamp
27001       S   0xB4  6    0      0  D->H V  I  0x04     0x4A10 0x0000 517       0x4B   5 . 791 366 082

    Packet  H H   SETUP ADDR ENDP CRC5 Pkt Len   Idle         Time Stamp
    97509     S   0xB4  6    0    0x09 10        267.330 ns   5 . 791 366 082

    Packet  H H   DATA0  Data                       CRC16  Pkt Len   Idle         Time Stamp
    97510     S   0xC3   C1 04 10 4A 00 00 05 02    0x74E6 16        267.330 ns   5 . 791 366 516

    Packet  D H   ACK    Pkt Len   Time        Time Stamp
    97511     S   0x4B   6         91.100 us   5 . 791 367 050
```

5. **Data Phase 2 (IN Data Payload Transaction):** Hub sends a series of IN packets (64 Bytes per packet) until all 517 bytes are read. The first 5 bytes should always be ignored.

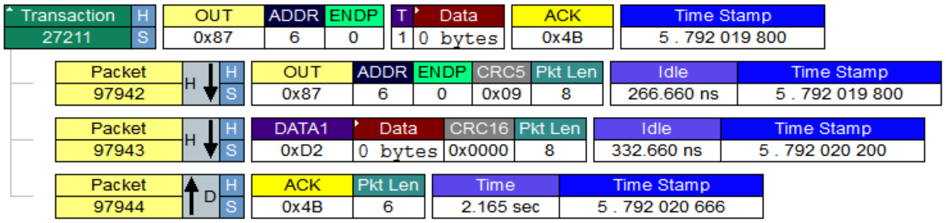
FIGURE 14: REGISTER READ IN TRANSACTION EXAMPLE



```
Transaction H  IN   ADDR ENDP  T  Data                                                                                                        ACK   Time       Time Stamp
27028       S  0x96 6    0     1   0: 00 00 00 00 00 01 00 02 48 BE 75 C8 10 22 FF 02 4F C5 90 09 54 E4 F0 30 25 04 E0 44 02 F0 22 02   0x4B  78.466 us  5 . 791 458 150
                                  32: 6D 39 02 46 A0 FF FF 02 67 38 FF FF FF FF FF 02 78 B0 90 08 0A E0 44 04 F0 78 BD E6 90 24 00 F0

Transaction H  IN   ADDR ENDP  T  Data                                                                                                        ACK   Time       Time Stamp
27051       S  0x96 6    0     0   0: 90 08 2F E0 44 30 F0 90 08 2E E0 44 02 F0 22 02 20 78 8B 2D 8A 2E 89 2F 8C 30 8D 31 C2 2C 75 34   0x4B  77.300 us  5 . 791 536 616
                                  32: 40 D3 E5 33 95 31 E5 32 95 30 50 08 85 32 30 85 33 31 80 12 E5 34 FD 7C 00 AE 30 AF 31 12 11 70

Transaction H  IN   ADDR ENDP  T  Data                                                                                                        ACK   Time       Time Stamp
27076       S  0x96 6    0     1   0: ED 4C 70 02 D2 2C E5 31 45 30 70 03 30 2C 73 90 33 40 E0 30 E0 07 90 33 43 E0 30 E0 F2 90 33 43   0x4B  77.400 us  5 . 791 613 916
                                  32: 74 01 F0 C3 E5 34 95 31 E4 95 30 AF 31 50 02 AF 34 8F 35 20 2B 28 E4 FF EF C3 95 35 50 20 AB 2D

Transaction H  IN   ADDR ENDP  T  Data                                                                                                        ACK   Time       Time Stamp
27103       S  0x96 6    0     0   0: 05 2F E5 2F AA 2E 70 02 05 2E 14 F9 12 10 F6 FE E4 2F F5 82 E4 34 33 F5 83 EE F0 0F 80 DA 90 33   0x4B  76.866 us  5 . 791 691 316
                                  32: 41 E5 35 F0 90 33 40 E0 44 01 F0 E5 31 45 30 70 02 C2 2C C3 E5 31 95 35 F5 31 E5 30 94 00 F5 30

Transaction H  IN   ADDR ENDP  T  Data                                                                                                        ACK   Time       Time Stamp
27128       S  0x96 6    0     1   0: 80 84 22 FF 75 81 BD 12 35 00 90 24 00 E0 30 E7 51 90 44 00 74 90 F0 A3 74 24 F0 A3 74 00 F0 A3   0x4B  76.400 us  5 . 791 768 182
                                  32: 74 E0 F0 A3 74 54 F0 A3 74 F7 F0 A3 74 F0 F0 A3 74 90 F0 A3 74 24 F0 A3 74 03 F0 A3 74 74 F0 A3

Transaction H  IN   ADDR ENDP  T  Data                                                                                                        ACK   Time       Time Stamp
27153       S  0x96 6    0     0   0: 74 0B F0 A3 74 F0 F0 A3 74 22 F0 90 08 42 74 10 F0 74 01 A3 F0 12 80 00 90 08 42 74 88 F0 A3 74   0x4B  79.184 us  5 . 791 844 582
                                  32: 81 F0 90 24 00 E0 44 02 F0 75 F9 98 00 00 12 6D 74 12 69 B7 02 7C 8F 02 7C 8F D2 2C 78 82 E6 78

Transaction H  IN   ADDR ENDP  T  Data                                                                                                        ACK   Time       Time Stamp
27180       S  0x96 6    0     1   0: B8 F6 E6 12 13 62 02 9B 00 03 82 01 03 74 02 02 9B 03 02 C3 04 03 5D 05 02 0C 06 02 17 07 02 27   0x4B  77.300 us  5 . 791 923 766
                                  32: 08 01 FB 09 02 A9 10 04 1A 20 04 49 21 04 78 22 04 BC 23 05 0C 24 05 AE 25 05 62 26 05 C1 28 07

Transaction H  IN   ADDR ENDP  T  Data                                                                                                        ACK   Time       Time Stamp
27205       S  0x96 6    0     0   0: 19 2A 07 19 2B 07 6D 2C 02 A9 30 07 8D 31 07 91 32 02 A9 40 02 A9 41 02 CD 42 02 A9 50 02 75 60   0x4B  12.666 us  5 . 792 001 066
                                  32: 02 7F 61 02 7A 62 02 A9 63 02 51 70 02 A9 71 03 D0 72 02 56 73 02 5B 74 07 95 80 00 00 07 B4 12

Transaction H  IN   ADDR ENDP  T  Data             ACK   Time       Time Stamp
27209       S  0x96 6    0     1  73 92 79 B0 7A   0x4B  3.434 us   5 . 792 013 732
```

6. **Status Phase 2 (OUT Transaction 2):** Host sends an OUT packet followed by a zero length data packet. The Hub Feature Controller ACKs to complete the bridging command.

FIGURE 15: REGISTER READ OUT TRANSACTION EXAMPLE



```
Transaction H | OUT  ADDR ENDP | T Data    | ACK  | Time Stamp
27211       S | 0x87 6    0    | 1 0 bytes | 0x4B | 5 . 792 019 800

  Packet H ↓ H | OUT  ADDR ENDP CRC5 Pkt Len | Idle       | Time Stamp
  97942      S | 0x87 6    0    0x09 8       | 266.660 ns | 5 . 792 019 800

  Packet H ↓ H | DATA1 Data    CRC16  Pkt Len | Idle       | Time Stamp
  97943      S | 0xD2  0 bytes 0x0000 8       | 332.660 ns | 5 . 792 020 200

  Packet ↑ D H | ACK  Pkt Len | Time      | Time Stamp
  97944      S | 0x4B 6       | 2.165 sec | 5 . 792 020 666
```

7. After completing the SPI Read, do one of the following:
   - Close the SPI Interface with the Disable SPI Pass-Through command.
   - Wait for the manufacturer specified time before performing another Read/Write command.
   - Send RDSR commands until the BUSY field is cleared before performing another Read/Write command.

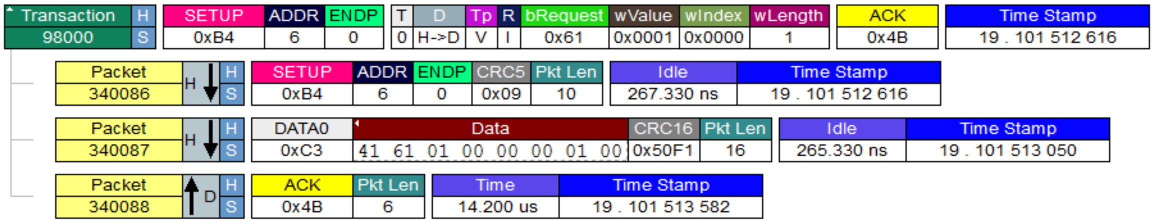
6.4 Write 256 Bytes to an attached SPI Device

1. **Command Phase 1 (SETUP Transaction 1):** This example shows how to perform a block write of 256 bytes to an attached SPI device. Send the following SETUP Register Read Command to Endpoint 0 of the Hub Feature Controller to enable the SPI write feature.

TABLE 13: SPI WRITE ENABLE SETUP PACKET EXAMPLE

Field	Value	Note
bmRequestType	0x41	—
bRequest	0x61	—
wValue	0x0001	—
wIndex	0x0000	—
wLength	0x0001	—

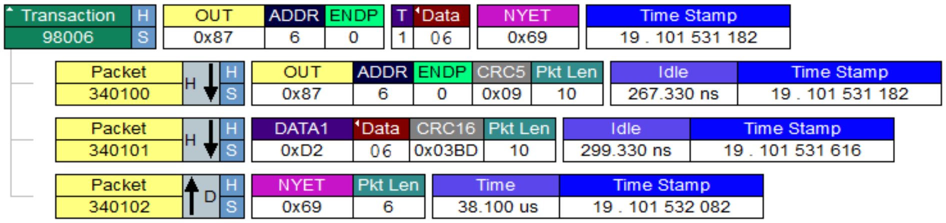
FIGURE 16: SPI WRITE ENABLE SETUP TRANSACTION EXAMPLE



```
Transaction H | SETUP ADDR ENDP | T D    Tp R bRequest wValue wIndex wLength | ACK  | Time Stamp
98000       S | 0xB4  6    0    | 0 H->D V  I 0x61     0x0001 0x0000 1       | 0x4B | 19 . 101 512 616

  Packet H ↓ H | SETUP ADDR ENDP CRC5 Pkt Len | Idle       | Time Stamp
  340086     S | 0xB4  6    0    0x09 10      | 267.330 ns | 19 . 101 512 616

  Packet H ↓ H | DATA0 Data                    CRC16  Pkt Len | Idle       | Time Stamp
  340087     S | 0xC3  41 61 01 00 00 00 01 00 0x50F1 16      | 265.330 ns | 19 . 101 513 050

  Packet ↑ D H | ACK  Pkt Len | Time      | Time Stamp
  340088     S | 0x4B 6       | 14.200 us | 19 . 101 513 582
```

2. **Data Phase 1 (OUT Transaction 1):** Host sends an OUT packet followed by a single byte data payload of 0x06. 0x06 is the SPI write enable command. The Hub Feature Controller responds with a NYET after receiving the data.

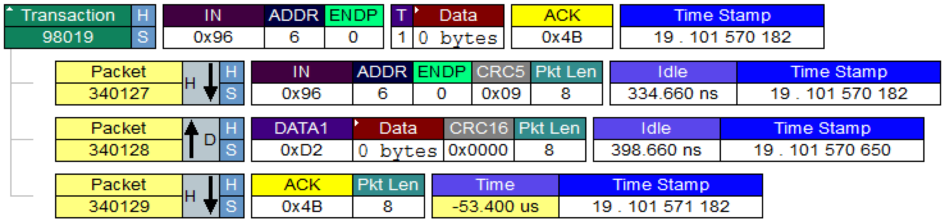
FIGURE 17: SPI WRITE ENABLE TRANSACTION EXAMPLE



```
Transaction H | OUT  ADDR ENDP | T Data | NYET | Time Stamp
98006       S | 0x87 6    0    | 1 06   | 0x69 | 19 . 101 531 182

  Packet H ↓ H | OUT  ADDR ENDP CRC5 Pkt Len | Idle       | Time Stamp
  340100     S | 0x87 6    0    0x09 10      | 267.330 ns | 19 . 101 531 182

  Packet H ↓ H | DATA1 Data CRC16  Pkt Len | Idle       | Time Stamp
  340101     S | 0xD2  06   0x03BD 10      | 299.330 ns | 19 . 101 531 616

  Packet ↑ D H | NYET Pkt Len | Time      | Time Stamp
  340102     S | 0x69 6       | 38.100 us | 19 . 101 532 082
```

3. **Status Phase 1 (IN Transaction 1):** Host sends an IN packet to complete the USB Transfer. Hub Feature Controller responds with a zero length data packet. The host ACKs to complete the bridging command.

FIGURE 18: SPI WRITE DATA OUT TRANSACTION EXAMPLE



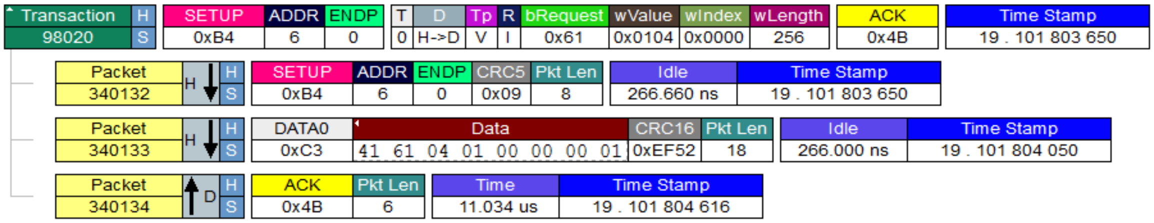
```
Transaction H | IN   ADDR ENDP | T Data    | ACK  | Time Stamp
98019       S | 0x96 6    0    | 1 0 bytes | 0x4B | 19 . 101 570 182

  Packet H↓ H | IN   ADDR ENDP CRC5 Pkt Len | Idle       | Time Stamp
  340127    S | 0x96 6    0    0x09 8       | 334.660 ns | 19 . 101 570 182

  Packet ↑D H | DATA1 Data    CRC16  Pkt Len | Idle       | Time Stamp
  340128    S | 0xD2  0 bytes 0x0000 8       | 398.660 ns | 19 . 101 570 650

  Packet H↓ H | ACK  Pkt Len | Time       | Time Stamp
  340129    S | 0x4B 8       | -53.400 us | 19 . 101 571 182
```

4. **Send the SPI Write Data Payload (SETUP Transaction 2):** The SPI device is now ready to receive the data payload. For a SPI Block Write of 256 bytes, the SETUP command is::

TABLE 14: SPI WRITE DATA SETUP PACKET EXAMPLE

Field	Value	Note
bmRequestType	0x41	—
bRequest	0x61	—
wValue	0x0104 (260)	The 256-byte data payload + 4 extra command bytes
wIndex	0x0000	—
wLength	0x0100 (256)	The 256-byte data payload

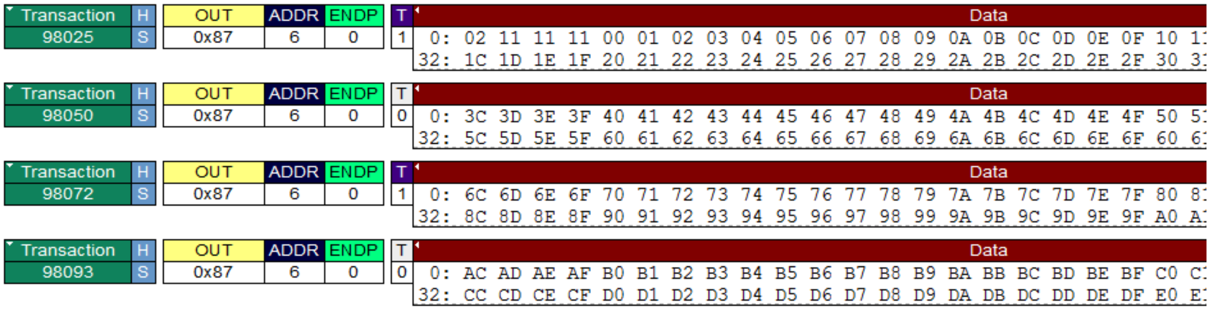
FIGURE 19: SPI WRITE DATA TRANSACTION EXAMPLE



```
Transaction H | SETUP ADDR ENDP | T D    Tp R bRequest wValue wIndex wLength | ACK  | Time Stamp
98020       S | 0xB4  6    0    | 0 H->D V  I 0x61     0x0104 0x0000 256     | 0x4B | 19 . 101 803 650

  Packet H↓ H | SETUP ADDR ENDP CRC5 Pkt Len | Idle       | Time Stamp
  340132    S | 0xB4  6    0    0x09 8       | 266.660 ns | 19 . 101 803 650

  Packet H↓ H | DATA0 Data                    CRC16  Pkt Len | Idle       | Time Stamp
  340133    S | 0xC3  41 61 04 01 00 00 00 01 0xEF52 18      | 266.000 ns | 19 . 101 804 050

  Packet ↑D H | ACK  Pkt Len | Time      | Time Stamp
  340134    S | 0x4B 6       | 11.034 us | 19 . 101 804 616
```

5. **Data Phase 2 (OUT Data Payload Transaction):** Host sends a series of OUT packets (64 Bytes per packet) until all 256 + 4 command bytes are sent. The first 4 bytes must be 0x02, 0xXX, 0xYY, 0xZZ where 0xXX, 0xYY, 0xZZ is the 24 bit physical address of the SPI Flash. In this example, the SPI address is 0x111111.

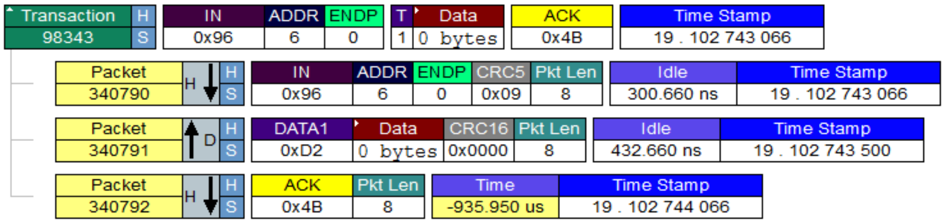
FIGURE 20: SPI WRITE DATA OUT TRANSACTIONS EXAMPLE



```
Transaction H | OUT  ADDR ENDP | T Data
98025       S | 0x87 6    0    | 1  0: 02 11 11 11 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 1
                                   32: 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 3

Transaction H | OUT  ADDR ENDP | T Data
98050       S | 0x87 6    0    | 0  0: 3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 5
                                   32: 5C 5D 5E 5F 60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 60 6

Transaction H | OUT  ADDR ENDP | T Data
98072       S | 0x87 6    0    | 1  0: 6C 6D 6E 6F 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F 80 8
                                   32: 8C 8D 8E 8F 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F A0 A

Transaction H | OUT  ADDR ENDP | T Data
98093       S | 0x87 6    0    | 0  0: AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF C0 C
                                   32: CC CD CE CF D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF E0 E
```

6. **Status Phase 2 (IN Transaction 2):** Host sends an IN packet and the hub responds with a zero length data packet. The Host ACKs to complete the bridging command.

FIGURE 21: SPI WRITE DATA IN TRANSACTION EXAMPLE



Transaction	H	IN	ADDR	ENDP	T	Data	ACK	Time Stamp
98343	S	0x96	6	0	1	0 bytes	0x4B	19 . 102 743 066

Packet		H	IN	ADDR	ENDP	CRC5	Pkt Len	Idle	Time Stamp
340790	H ↓	S	0x96	6	0	0x09	8	300.660 ns	19 . 102 743 066

Packet		H	DATA1	Data	CRC16	Pkt Len	Idle	Time Stamp
340791	↑ D	S	0xD2	0 bytes	0x0000	8	432.660 ns	19 . 102 743 500

Packet		H	ACK	Pkt Len	Time	Time Stamp
340792	H ↓	S	0x4B	8	-935.950 us	19 . 102 744 066

7. After completing the SPI Write, do one of the following:
   - Close the SPI Interface with the Disable SPI Pass-Through command.
   - Wait for the manufacturer specified time before performing another Read/Write command.
   - Send RDSR commands until the BUSY field is cleared before performing another Read/Write command.

APPENDIX A: APPLICATION NOTE REVISION HISTORY

TABLE A-1: REVISION HISTORY

Revision Level & Date	Section/Figure/ Entry	Correction
DS00002430A (mm-dd-17)	All	Initial release.

NOTES:

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BeaconThings, BitCloud, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, RightTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, CryptoAuthentication, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, QMatrix, RightTouch logo, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-5224-1661-6

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-3326-8000
Fax: 86-21-3326-8021

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

France - Saint Cloud
Tel: 33-1-30-60-70-00

Germany - Garching
Tel: 49-8931-9700
Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-67-3636

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7289-7561

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

11/07/16